



Architettura del software: una introduzione

Andrea Saltarello

Software Architect @ Managed Designs S.r.l.

andrea.saltarello@manageddesigns.it

<http://blogs.ugidotnet.org/pape>



GUISA

Gruppo Utenti Italiani
Solution Architect



<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Chi sono

- Solution Architect @ **Managed Designs S.r.l.**
(<http://www.manageddesigns.it>)
- Presidente dello **User Group Italiano .NET**
(**UGIdotNET** – <http://www.ugidotnet.org>)
- Fondatore del **Gruppo Italiano Utenti Solution Architect** (**GUISA** – <http://www.guisa.org>)

Serie webcast "Architettura del software"

- 6/11/2006 – una introduzione
- 17/11/2006 - dai requisiti ai casi d'uso
- 22/11/2006 - dai casi d'uso al modello
- 27/11/2006 - un approccio agile
- 30/11/2006 - introduzione ai design pattern
- 06/12/2006 - patterns by example
- 13/12/2006 - disegno architeturale, gli idiomi e le linee guida di design per il .NET Framework
- 20/12/2006 - .NET e gli strumenti

Cosa è l'Architettura?

Secondo la definizione ANSI/IEEE Std
1471-2000:

"L'organizzazione basilare di un sistema, rappresentato dalle sue componenti, dalle relazioni che esistono tra di loro e con l'ambiente circostante, e dai principi che governano la sua progettazione e evoluzione."

Architettura: cosa?

L'architettura di un sistema **è** definita durante la fase di progettazione.

L'architettura **non è** parte dell'analisi:

- l'analisi si concentra su *cosa* debba fare il sistema
- La progettazione si concentra su *come* debba farlo

Architetto: le responsabilità

L'architetto:

- Recepisce i requisiti *funzionali* (emersi durante l'analisi) e quelli *non funzionali* (es: HA, scalabilità, security ...)
- Effettua una analisi del rapporto costi/benefici per determinare il miglior modo di soddisfare i requisiti, eventualmente ricorrendo a componenti commerciali o comunque già sviluppati
- Suddivide i grandi sistemi in (*layer* di) sottosistemi individualmente assegnabili ad uno sviluppatore o ad un gruppo di lavoro
- Genera "specifiche": *sketch*, modelli o prototipi per comunicare al gruppo di lavoro le modalità di realizzazione

Responsabilità dell'architetto: Requisiti

L'architetto recepisce i requisiti *funzionali* (emersi durante l'analisi) e quelli *non funzionali* (es: HA, scalabilità, security ...)

- 17/11/2006 - dai requisiti ai casi d'uso
- 27/11/2006 - un approccio agile
- 20/12/2006 - .NET e gli strumenti

Responsabilità dell'architetto: le Strategie

L'architetto effettua una analisi del rapporto costi/benefici per determinare il miglior modo di soddisfare i requisiti, eventualmente ricorrendo a componenti commerciali o comunque già sviluppati

- 17/11/2006 - dai requisiti ai casi d'uso
- 22/11/2006 - dai casi d'uso al modello
- 30/11/2006 - introduzione ai design pattern
- 06/12/2006 - patterns by example
- 13/12/2006 - disegno architettuale, gli idiomi e le linee guida di design per il .NET Framework

Responsabilità dell'architetto: il Partitioning

L'architetto suddivide i grandi sistemi in (*layer* di) sottosistemi individualmente assegnabili ad uno sviluppatore o ad un gruppo di lavoro

- 22/11/2006 - dai casi d'uso al modello
- 27/11/2006 - un approccio agile
- 30/11/2006 - introduzione ai design pattern
- 06/12/2006 - patterns by example
- 13/12/2006 - disegno architettuale, gli idiomi e le linee guida di design per il .NET Framework
- 20/12/2006 - .NET e gli strumenti

Responsabilità dell'architetto: la Comunicazione

L'architetto genera "specifiche": abbozzi (*sketch*), modelli o prototipi per comunicare al gruppo di lavoro le modalità di realizzazione

- 17/11/2006 - dai requisiti ai casi d'uso
- 22/11/2006 - dai casi d'uso al modello
- 27/11/2006 - un approccio agile
- 30/11/2006 - introduzione ai design pattern
- 06/12/2006 - patterns by example
- 13/12/2006 - disegno architettuale, gli idiomi e le linee guida di design per il .NET Framework
- 20/12/2006 - .NET e gli strumenti

*Determining what the users/sponsors actually want, rather than what they say they want, is not engineering— it is an art. An architect does not follow an exact procedure. S/he communicates with users/sponsors in a highly interactive way— together they extract the true requirements necessary for the engineered system. The software architect must remain constantly in communication with the end users. Therefore, the architect must be familiar with the user's environment and problem.
(cit.)*

Architettura: luoghi comuni

- Architetto \neq Analista
- Architetto \neq Project Manager
- Architettura \neq Design
 - Al limite... Design \in Architettura

Architettura != Design

Un esempio reale: **GazzaTown**
[<http://gazzatown.gazzetta.it>]

La Gazzetta dello Sport



WEB VAS
K SOLUTION
INFRASTRUTTURE

managed/designs



GazzaTown

Caratteristiche:

- Pipeline e-commerce gestita da **Commerce Server 2007**
- Integrazione con back-end gestita con **BizTalk 2006**

Pressochè totale assenza di design. In veste di architetto, ho:

- Recepto i requisiti
- Scelto i sistemi da utilizzare
- Specificato gli schemi XSD dei flussi dati
- Scritto parte del codice per l'interfacciamento al *gateway* di Banca Sella ☺

Design € Architettura

Progetto NSK:

- Raccolta dei requisiti
- Design *up-front* (non troppo big 😊)

<http://download.manageddesigns.it/nsk.aspx>

Bibliografia

ANSI/IEEE:

- http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html

Wikipedia:

- http://en.wikipedia.org/wiki/Software_architect
- http://en.wikipedia.org/wiki/Ieee_1471

Link

<http://www.guisa.org>

<http://www.guisa.org/forums>

Dubbi/Domande/Curiosità? 😊